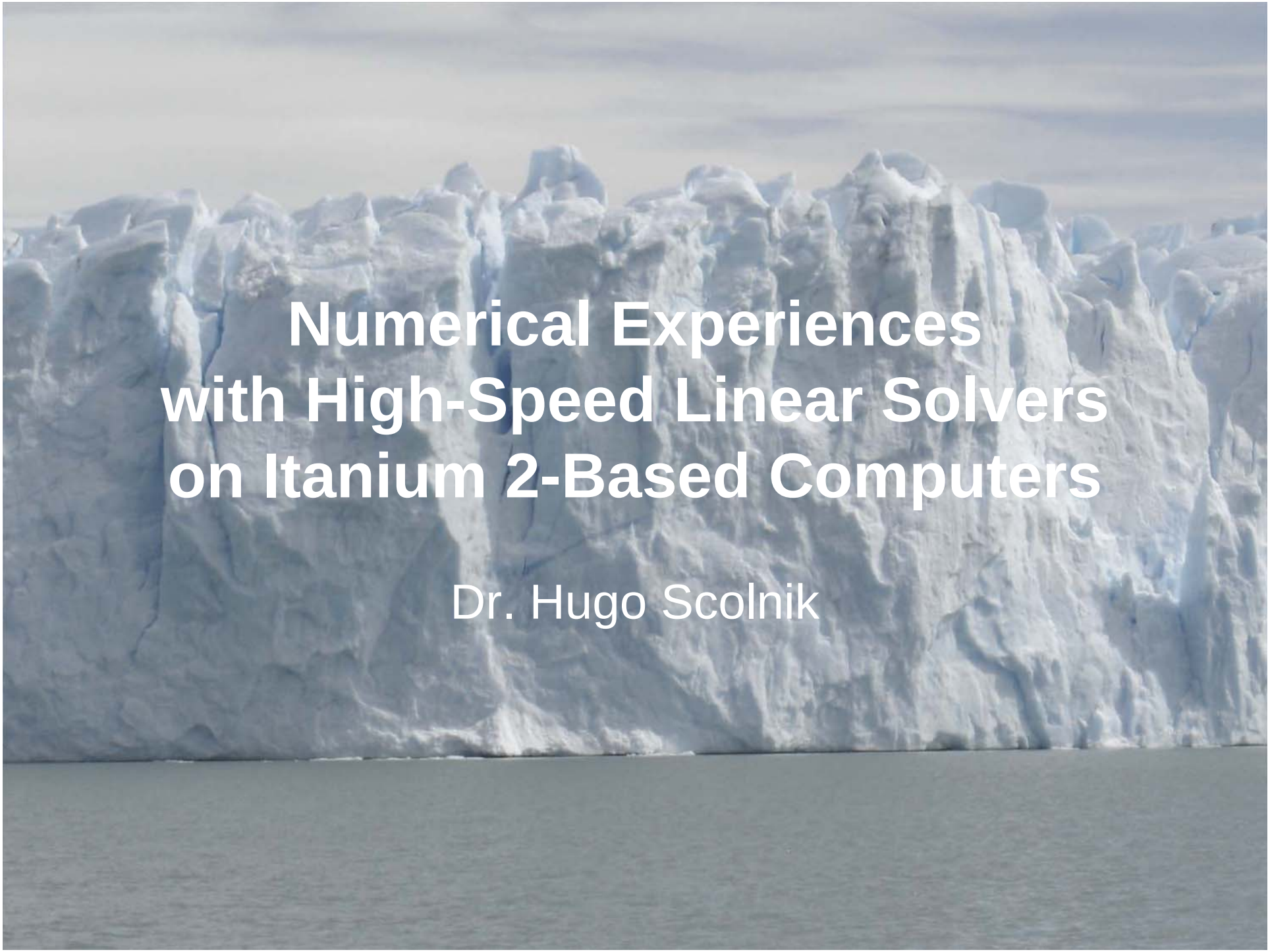
Department of Computer Science

School of Sciences

Universidad de Buenos Aires

Argentina



Numerical Experiences with High-Speed Linear Solvers on Itanium 2-Based Computers

Dr. Hugo Scolnik

About Us

- Largest computer science department in the country.
- 150 total staff, 33 professors (19 full time).
- Our students rated 10th in ACM Programming Contest 2002, above all other North & South American teams (and 14th in 2007).
- Most Computer Science international publications from Argentina come from our Department.
- Website: www.dc.uba.ar

CS Department: Joint Activities

- Pontifícia Universidade Católica de Rio de Janeiro, Brazil
- UNICAMP, Brazil
- Universidade Federal de Rio de Janeiro, Brazil
- Universidade Federal de Pernambuco, Brazil
- Universidade Federal de Porto Alegre, Brazil
- ZIB-Berlín, Germany
- Università di Roma “La Sapienza”, Italy
- Università di Pisa, Italy
- University of Toronto
- King’s College, University of London
- IEI-CNR, Pisa, Italy
- Politécnico de Milano, Italy
- University of Haifa, Israel
- University of Pennsylvania
- CNRS, Grenoble, France
- Ècole Nationale Superieur des Télécommunications, Paris, France
- Dep.of Biometry, Maison Nationale des Éleveurs, Paris, France
- Drexel University, Pennsylvania
- École Polytechnique et CNRS, France
- SRI International, California
- DECSAI, University of Granada, Spain
- South Bank University, London, England
- Polytechnic University, Barcelona, Spain
- German Aerospace Agency, Baviera
- IIIA-CSIC, Barcelona, Spain
- LRI of the Université Orsai, Paris Sud, France
- Department of Mathematics, Université Paris 7, France
- Department of Mathematics, University of Siena, Italy.
- IBM Watson Research Center, New York
- Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, New Zealand
- IRIT-CNS, Toulouse, France
- Oak-Ridge National Laboratory, TN
- Université de Aix-Marseille, France
- All national universities in Argentina and most in Chile, and Uruguay

Itanium Equipment

Hardware

HP Integrity rx2600

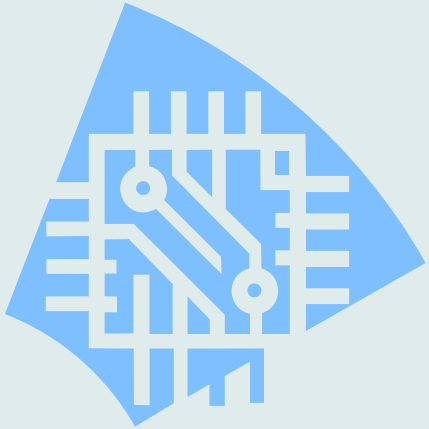
Processors: 2x Intel Itanium 2

Clock frequency: 1.5 GHz

Cache: 6MB L3

System bus bandwidth: 6.4 GB/s.

Memory: 2 GB DDR, 8.5 GB/sec bandwidth



Software

Operating System: Debian Linux Version 3.1r1 (Sarge)

Compiler: Intel C++ Compiler for Itanium-based applications Version 9.1.042 (20060707)

Math Libraries:

- Intel Math Kernel Library 9.0
- HP MLIB 9.2

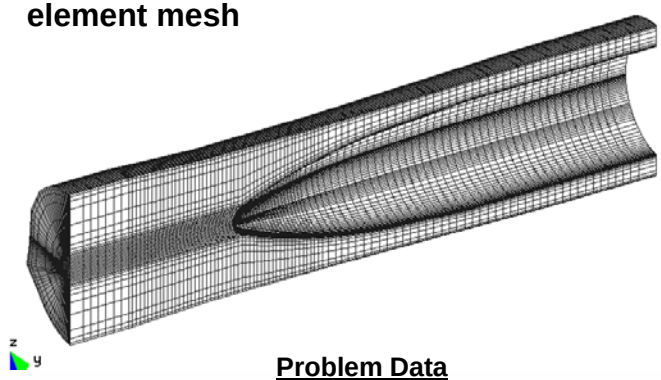
Current work

The problem of **solving very large systems** of linear algebraic equations appears in many real world applications.

The following problems arising from the use of the Finite Elements Method are being solved by SIDERCA, a company belonging to the TENARIS group, a worldwide manufacturer of seamless steel tubes for several industries.

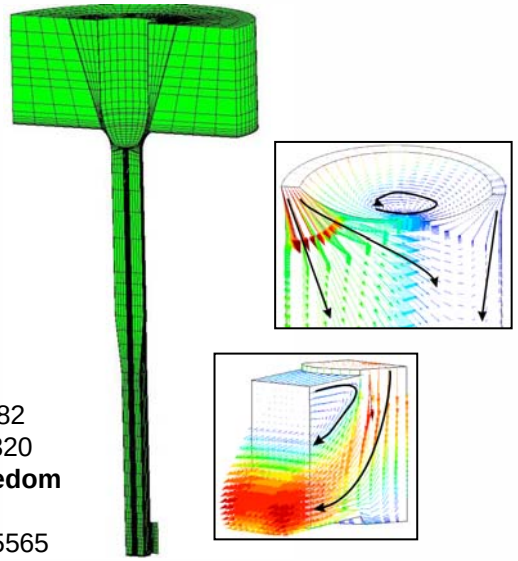
Finite element simulation of the Mannesmann piercing process

View of the finite element mesh



Problem Data
Dimensions : 3
Nodes: 109148
Elements: 100950
Number of unknowns 322894

Finite element modeling of control flow devices in steel industry

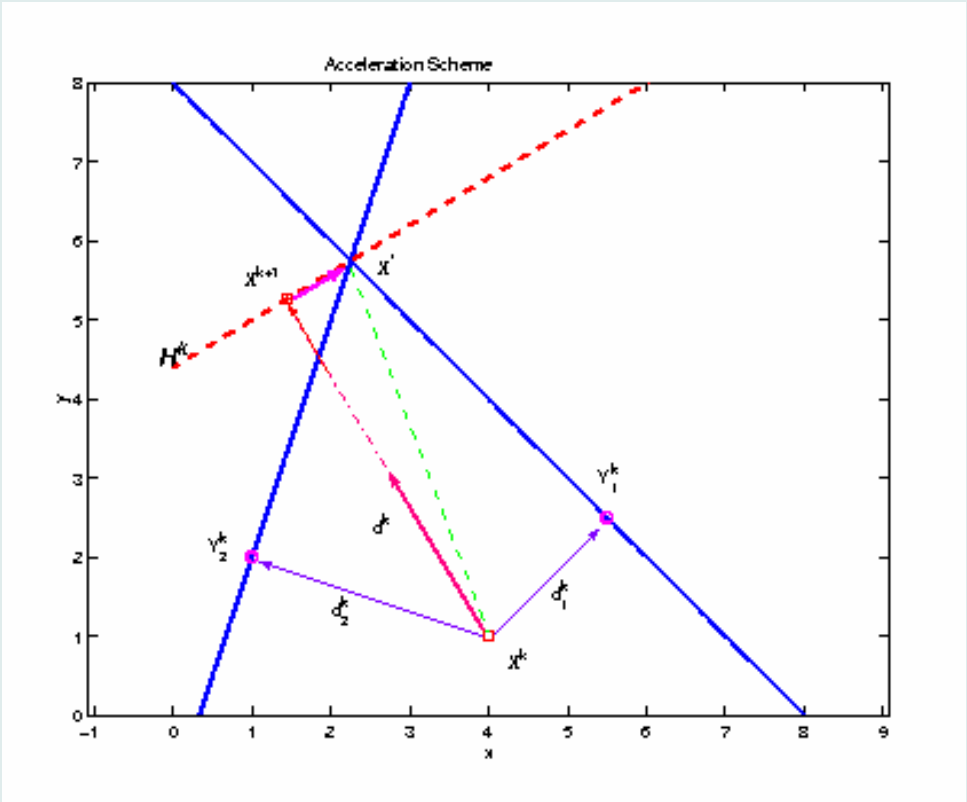


Problem Data
Dimensions: 3
Nodes: 84582
Elements: 76320
Degrees of freedom (Navier-Stokes equation): 235565

Beyond certain sizes, direct methods can not be applied to solve these problems and iterative methods are employed.

AcCim Algorithm

We have developed a new category of fast, intrinsically parallel iterative methods for solving very large systems of linear equations.



Publications

"New Optimized and Accelerated PAM Methods for Solving Large Non-Symmetric Systems", H. Scolnik, N.Echebest, M.T.Guardarucci, M.C. Vacchino.in the book: Inherently Parallel Algorithms in Feasibility and Optimization and their Applications, D.Butnariu, Y. Censor and S. Reich (Editors), Studies in Computational Mathematics 8, 2001 Elsevier Science, Amsterdam, pp 457-471,2001

"A Class of Optimized Row Projection Methods for Solving Large Non-Symmetric Linear Systems", H.Scolnik, N.Echebest, M.T.Guardarucci, M.C.Vacchino, Applied Numerical Mathematics 41, 4, pp.499-513, 2002, Elsevier Science.

"Acceleration Scheme for Parallel Projected Aggregation Methods for Solving Large Linear Systems", H.Scolnik, N.Echebest, M.T.Guardarucci, M.C.Vacchino. Annals of Operations Research. Volume 117, 2002, Baltzer Science Publishers.

"An acceleration scheme for solving convex feasibility problems using incomplete projection algorithms", N. Echebest, M. T. Guardarucci, H. D. Scolnik, M. C. Vacchino , Numerical Algorithms., 35, pp.331-350, 2004

"Incomplete Oblique Projections Algorithms for Solving Large Inconsistent Linear Systems", H.Scolnik, N.Echebest, M.T.Guardarucci, Mathematical Programming, 2007.

Solver Implementation

We developed and optimized a solver based on the AcCim algorithm.

- It is written in the C++ language and compiled with the Intel Compiler.
- It supports dense and symmetric and non-symmetric sparse general systems.
- We developed several implementations of the software, and tried different approaches to speed-up execution (compiler-based optimizations, multithreading using pthreads and OpenMP).
- The current version improved more than 200% over the previous one by using high level and algorithmic optimizations. We're planning on trying the Intel Threading Building Blocks.

Test Results with AcCim Solver

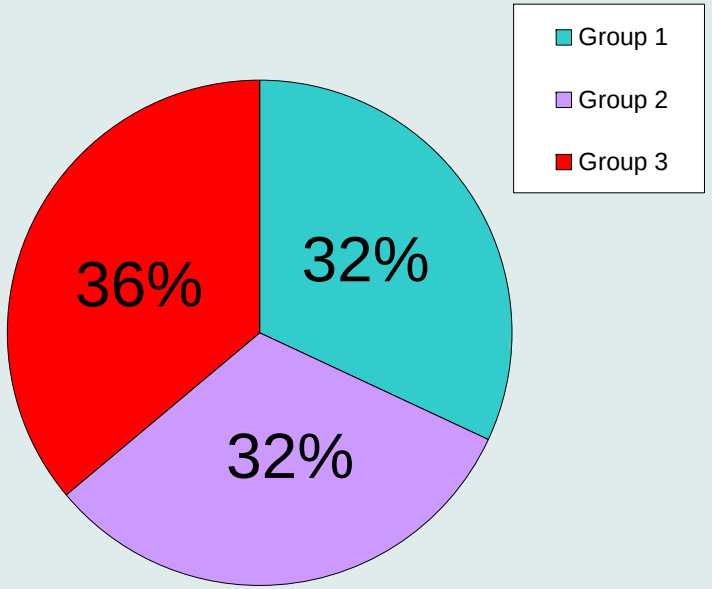
The last version of our solver was tested with 25 ill-conditioned systems from different problem domains.

We can classify the results into three groups:

- Group 1: The results obtained were good in both, time and precision.

- Group 2: The results obtained were good in precision, but needed too many iterations to converge.

- Group 3: The results were not precise enough.



Some good results: B&S matrices

We tested the AcCim solver with the six difficult problems from computational mechanics, proposed by Bramley and Sameh in [1].

These problems have sparse, non-symmetric matrices with 13824 rows and columns, and 93312 non-zero values.

	P1	P2	P3	P4	P5	P6
Estimated *k*(A)	9	57	40000	4786	36	77

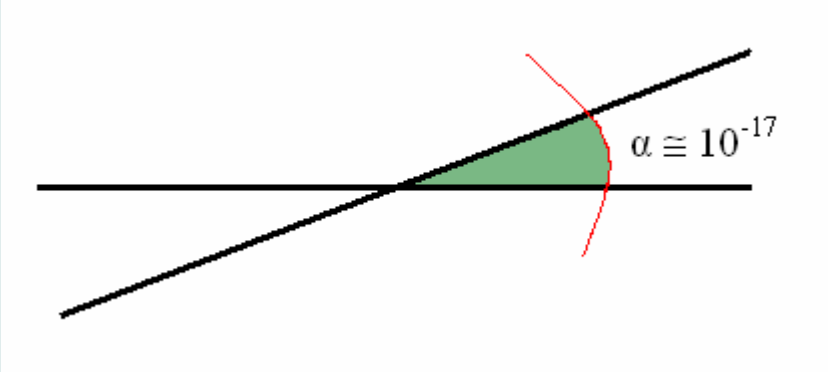
The following results were obtained running the AcCim solver on an Itanium 2 machine.

	P1	P2	P3	P4	P5	P6
Residual	9.68178e-08	1.95547e-05	0.000153996	6.11926e-06	2.52025e-06	5.73289e-06
$\|X^* - X_k\|$	2.76788e-07	5.73204e-05	0.00563879	5.90394e-05	9.06588e-06	8.0397e-06
Time (secs)	0.521184	1.84366	2.67326	7.45664	0.564128	0.466528

[1] R. Bramley and A. Sameh. Row projection methods for large nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 168-193

Analyzing non satisfactory results

We analyzed the geometry of all matrices, particularly where the result obtained was not satisfactory.
We found that the worst systems are those with a large number of almost parallel rows.



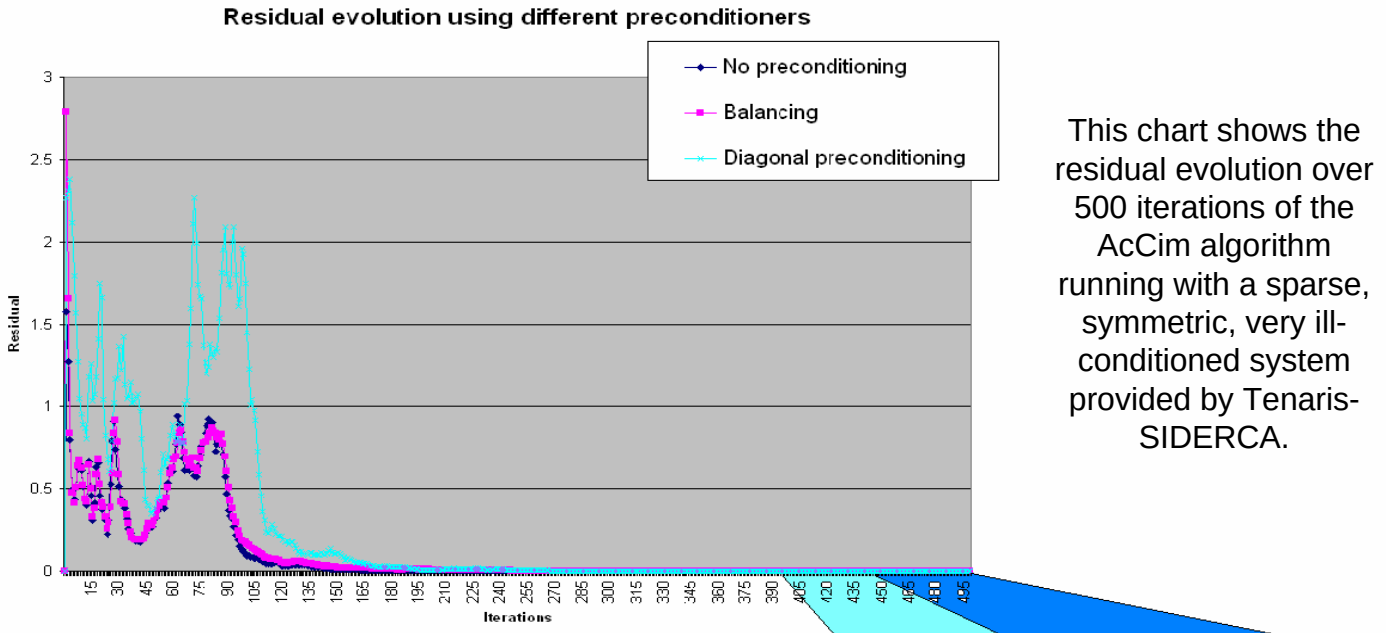
Preconditioners

To improve convergence with this kind of matrices we tested different preconditioners:

- inverse diagonal pre-multiplication
- column normalization
- row and column balancing

Preconditioners

The preconditioners tested so far haven't improved the convergence obtained by running AcCim solver on the original matrices. An example can be seen in the following chart:

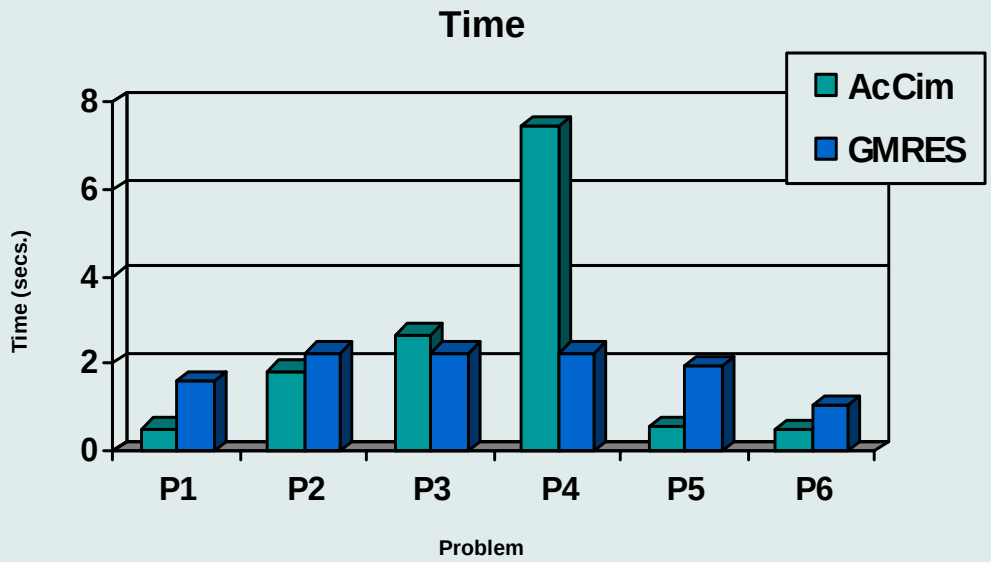


This chart shows the residual evolution over 500 iterations of the AcCim algorithm running with a sparse, symmetric, very ill-conditioned system provided by Tenaris-SIDERCA.

Problem size: 13345x13345

Number of non-zero values: 474966

Comparison with GMRES(*)



Precision (Final residual)

	P1	P2	P3	P4	P5	P6
AcCim	9.68178e-08	1.95547e-05	0.000153996	6.11926e-06	2.52025e-06	5.73289e-06
GMRES	2.16674e-006	0.0506362	0.0474483	108.796	0.000333852	0.000246743

(*) Open source version downloaded from http://people.scs.fsu.edu/~burkardt/cpp_src/mgmres/mgmres.html.

Future Work:

Our solver is very good with general and ill-conditioned matrices, but still presents some problems with some almost singular ill-conditioned matrices.

In order to find a solution for this problem, we are moving along two lines of work:

✓Implementation of a block-splitting algorithm. Although this is a time consuming solution, it can be reused for several right hand sides.

✓Development of a new kind of preconditioners. They have to preserve matrix sparsity as well as the necessary properties of the systems.

¡Gracias!

(Oh, sorry, in English: Thanks!)